

Testing Probabilistic Processes: Can Random Choices Be Unobservable?

Sonja Georgievska and Suzana Andova

Department of Mathematics and Computer Science, Eindhoven University of
Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
`s.georgievska@tue.nl`, `s.andova@tue.nl`

Abstract. A central paradigm behind process semantics based on observability and testing is that the exact moment of occurring of an internal nondeterministic choice is unobservable. It is natural, therefore, for this property to hold when the internal choice is quantified with probabilities. However, ever since probabilities have been introduced in process semantics, it has been a challenge to preserve the unobservability of the random choice, while not violating the other laws of process theory and probability theory. This paper addresses this problem. It proposes two semantics for processes where the internal nondeterminism has been quantified with probabilities. The first one is based on the notion of testing, i.e. interaction between the process and its environment. The second one, the probabilistic ready trace semantics, is based on the notion of observability. Both are shown to coincide. They are also preserved under the standard operators.

1 Introduction

A central paradigm behind process semantics based on observability (e.g. [11]) is that the exact moment of occurring of an internal nondeterministic choice is unobservable. This is because an observer does not have insight into the internal structure of a process but only in the externally visible actions. Unobservability of internal choice has been also accomplished by the testing theory [6]¹. It is natural, therefore, for this property to hold when the internal choice is quantified with probabilities. However, it turned out that unobservability of internal probabilistic choice is not trivial to achieve in probabilistic testing theory. To explain why, we start with an example.

Motivation Consider a machine which flips a fair coin internally. A user can guess the result of the flipping by pressing a “head” or a “tail” button. If the user has guessed correctly, the machine offers a prize. The machine can be modeled by process graph (or shortly process) s in Fig. 1 and the user can be modeled by process u in Fig. 1. The user is happy if, after pressing a button, a prize follows.

¹ In fact, the process semantics based on [11] and [6] do coincide for a broad class of processes, as shown in [19].

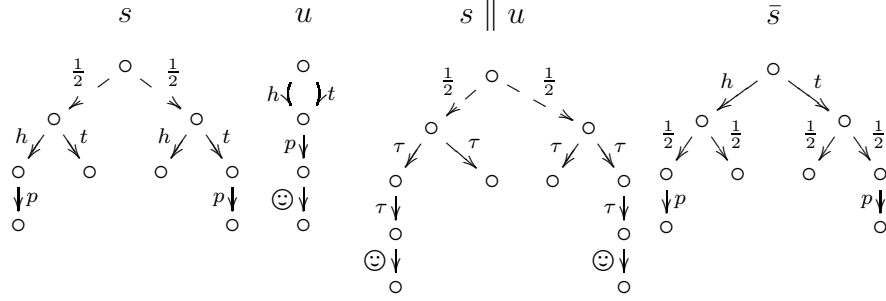


Fig. 1: Processes s and \bar{s} are distinguished in probabilistic may/must testing theory

Let the user and the machine interact, i.e. let them synchronize on all actions (except on the “user happiness” reporting action \odot). In terms of testing theory [6], process s is tested with test u . Intuitively, the probability that the user has guessed the output of flipping is $\frac{1}{2}$. That is, the probability of a \odot action being reported is $\frac{1}{2}$. However, most of the existing approaches for probabilistic testing, in particular probabilistic may/must testing [7, 12, 20, 23, 25], do not give this answer. Consider the synchronization $s \parallel u$ represented by the graph in Fig. 1, where actions are hidden after they have synchronized. In order to compute the probability of \odot being reported, the approaches in [7, 12, 20, 23, 25] use *schedulers*, that have insight into the internal structure of the graph of the synchronized system. Each scheduler resolves the nondeterminism in the nondeterministic nodes of $s \parallel u$ and yields a fully probabilistic system. For $s \parallel u$ in Fig. 1, there are four possible schedulers, which yield the following set of probabilities with which s passes the test u : $\{0, \frac{1}{2}, 1\}$. We can see that, because the power of the schedulers is unrestricted, unrealistic upper and lower bounds for the probabilities are obtained. Observe that this happens due to the effect of “cloning” the nondeterminism after hiding the synchronized actions. The choice between h and t has been “cloned” in both futures after the probabilistic choice in $s \parallel u$. When resolving nondeterminism in $s \parallel u$, a scheduler assumes that the user has unrealistic power to *see* the result of the coin-flipping *before* guessing.

The above example challenges us to reconsider the design choice to hide actions after synchronization. Namely, although hiding is harmless and actually useful in [6], and helps to abstract away from unnecessary information, in probabilistic testing it may actually “hide too much” and produce overestimation of the probability information about the system. It is highly undesirable to obtain lower and upper probability bounds of 0 and 1 resp. for the probabilistic behaviour of a simple system (as the one in Fig. 1), when the actual probability is $\frac{1}{2}$. This may render a testing equivalence insufficient for verification purposes.

Consider now process \bar{s} in Fig. 1. To the user this graph may as well represent the behaviour of the coin-flipping machine – the user cannot see whether the machine flips the coin *before* or *after* making the “head or tail” guess. According to her, the machine acts as specified as long as she is able to guess the result in half of the cases. In fact, both schedulers applied to $\bar{s} \parallel u$ yield that the

probability of reporting a \odot action is exactly $\frac{1}{2}$. Because of the last, none of the approaches in [7, 12, 20, 23, 25] equate processes s and \bar{s} , as, when tested with u , they produce different bounds for the probabilities of reporting \odot .² Note that being able to equate s and \bar{s} means allowing distribution of external choice over internal probabilistic choice [11].

Not allowing distribution of external choice over internal probabilistic choice has an additional effect, undesirable for compositional verification. Namely, if distribution of external choice over internal probabilistic choice is not allowed, then distribution of prefix over internal probabilistic choice is questioned too, as this implies congruence issues for *asynchronous* or *concurrent* parallel composition [11] (where processes synchronize on their common actions while interleave on the other actions). For instance, we would not be able to equate processes $e.a.(b \oplus_{\frac{1}{2}} c)$ and $e.((a.b) \oplus_{\frac{1}{2}} (a.c))$. (The operator “.” stands for prefixing and the operator “ \oplus ” stands for a probabilistic choice.) This is because these two processes, running each concurrently with process $e.d$, yield systems that cannot be equated, unless we allow distribution of external choice over internal probabilistic choice. If we are not able to relate processes $e.a.(b \oplus_{\frac{1}{2}} c)$ and $e.((a.b) \oplus_{\frac{1}{2}} (a.c))$, i.e. to allow distribution of prefix over internal probabilistic choice, then for verification we can only rely on equivalences that inspect the internal structure of processes, as bisimulations and simulations [10], and, moreover, expect overestimation of probabilities.

All together, the above discussions trigger the following question: “In a model where the internal nondeterminism has been quantified with probabilities [14], is it possible to test process s with test u (Fig. 1) such that the result of testing would imply that the probability of s passing the test u is exactly $\frac{1}{2}$?” In this case not only we could preserve the information on probability, but we could also allow distribution of prefix over probabilistic choice without losing compositionality.

Contributions In this paper we show that the answer to the above question is positive. The main contributions of the paper are the following:

- We introduce a technique for labeling the synchronized actions when a reactive probabilistic process is tested (Section 3). The labels are in form of rational functions, whose argument names are constructed from the action labels set. The labeling is achieved automatically when processes synchronize, i.e. no additional manipulation on the process graphs is needed.
- We propose a testing semantics (Section 3) exploiting the new labeling method, such that the result of testing process s with test u in Fig. 1 is $\frac{1}{2}$, and processes s and \bar{s} in Fig. 1 are testing-equivalent.
- We define a probabilistic ready trace equivalence for reactive probabilistic processes using the Bayesian definition of probability (Section 4). The definition allows a testing scenario in the lines of [4, 10] to be easily constructed.
- We define an algebra of finite processes and show that the ready trace equivalence is congruence for the standard operators (Section 5).

² If we ignore the probabilities, processes s and \bar{s} are testing-equivalent by [6].

- We show that all operators of our algebra, including external choice, distribute over probabilistic choice, allowing us to consider the latter one as unobservable (Section 5).
- We show that the testing equivalence of Sec. 3 and the ready-trace equivalence of Sec. 4 coincide (Section 6).

Section 7 ends with concluding remarks, future work directions regarding coexistence of probabilistic and internal choice, and related work.

2 Preliminaries

We define some preliminary notions needed for the rest of the paper.

Bayesian probability For a set A , 2^A denotes its power-set. The following definitions are taken from [15].

We consider a sample space, Ω , consisting of points called *elementary events*. Selection of a particular $a \in \Omega$ is referred to as an “ a has occurred”. An *event* is a set of elementary events. A, B, C, \dots range over events. An event A *has occurred* iff for some $a \in A$ a has occurred. Let A_1, A_2, \dots be a sequence of events and C be an event. The members of the sequence are *exclusive given C* , if whenever C has occurred no two of them can occur together, that is, if $A_i \cap A_j \cap C = \emptyset$ whenever $i \neq j$. C is called a *conditioning* event. If the conditioning event is Ω , then “given Ω ” is omitted.

For certain pairs of events A and B , a real number $P(A|B)$ is defined and called the *probability* of A given B . These numbers satisfy the following axioms:

A1: $0 \leq P(A|B) \leq 1$ and $P(A|A) = 1$.

A2: If the events in $\{A_i\}_{i=1}^{\infty}$ are exclusive given B , then $P(\cup_{i=1}^{\infty} A_i \mid B) = \sum_{i=1}^{\infty} P(A_i|B)$.

A3: $P(C|A \cap B) \cdot P(A|B) = P(A \cap C|B)$.

For $P(A|\Omega)$ we simply write $P(A)$.

Probabilistic transition systems In a probabilistic transition system (PTS) there are two types of transitions, viz. action and probabilistic transitions; a state can either perform action transitions only (nondeterministic state) or (unobservable) probabilistic transitions only (probabilistic state). To simplify, we assume that probabilistic transitions lead to nondeterministic states. The nondeterministic states exhibit only a so-called *external* (observable) nondeterminism, i.e the choice is between the actions, but once the action is chosen, the next state is determined. The outgoing transitions of a probabilistic state s define probability over the power-set of the set of nondeterministic states.

We give a formal definition of a PTS. Presuppose a finite set of actions \mathcal{A} .

Definition 1 (Probabilistic Transition System (PTS)). A PTS is a tuple $\mathcal{P} = (S_n, S_p, \rightarrow, \dashrightarrow)$, where

- S_n and S_p are finite disjoint sets of nondeterministic and probabilistic states, resp.,
- $\rightarrow \subseteq S_n \times \mathcal{A} \times S_n \cup S_p$ is an action transition relation such that $(s, a, t) \in \rightarrow$ and $(s, a, t') \in \rightarrow$ implies $t = t'$, and
- $\dashrightarrow \subseteq S_p \times (0, 1] \times S_n$ is a probabilistic transition relation such that, for all $s \in S_p$, $\sum_{(s, \pi, t) \in \dashrightarrow} \pi = 1$.

We denote $S_n \cup S_p$ by S . We write $s \xrightarrow{a} t$ rather than $(s, a, t) \in \rightarrow$, and $s \xrightarrow{\pi} t$ rather than $(s, \pi, t) \in \dashrightarrow$ (or $s \dashrightarrow t$ if the value of π is irrelevant in the context). We write $s \xrightarrow{a}$ to denote that there exists an action transition $s \xrightarrow{a} s'$ for some $s' \in S$. We agree that a state without outgoing transitions belongs to S_n .

As standard, we define a *process graph* (or simply *process*) to be a state $s \in S$ together with all states reachable from s , and the transitions between them. A process graph is usually named by its *root* state, in this case s .

3 Testing equivalence

In this section we define a testing equivalence in the style of [6] for reactive probabilistic processes.

Recall from elementary mathematics that a division of two polynomials is called a *rational function*. For example, $\frac{2x}{x+y}$ is a rational function with arguments x and y . A possible domain for this function is $(0, \infty) \times (0, \infty)$. We are going to exploit a subset \mathcal{R} of the rational functions whose argument names belong to the action labels \mathcal{A} , which is generated by the following grammar:

$$\varphi ::= \alpha \mid a \mid \varphi + \varphi \mid \varphi \cdot \varphi \mid \frac{\varphi}{\varphi},$$

where α is a non-negative scalar, $a \in \mathcal{A}$, and $+$, \cdot , and \div are ordinary algebraic addition, multiplication and fraction, resp. Brackets are used in the standard way to change the priority of the operators. For our purposes, we assume that the arguments a, b, \dots can only take positive values, i.e. the domain of every function in \mathcal{R} is $(0, \infty)^n$, where n is the size of the action set. Therefore, two rational functions in \mathcal{R} are equal iff they can be transformed to equal terms using the standard transformations that preserve equivalence (e.g. for $a, b \in \mathcal{A}$, $\frac{1}{2} \cdot \frac{a}{a+b} + \frac{1}{2} \cdot \frac{b}{a+b} = \frac{1 \cdot (a+b)}{2 \cdot (a+b)} = \frac{1}{2}$).

As standard, a *test* T is a finite process such that, for a symbol $\omega \notin \mathcal{A}$, there may exist transitions $s \xrightarrow{\omega}$ for some states s of T . Denote the set of all tests by \mathcal{T} . Given a process s and action $a \in \mathcal{A}$, denote by s_a the process (if exists) for which $s \xrightarrow{a} s_a$. Given a PTS $\mathcal{P} = (S_n, S_p, \rightarrow, \dashrightarrow)$, let $I: S_n \mapsto 2^{\mathcal{A}}$ be a function such that, for all $a \in \mathcal{A}$, $s \in S_n$, it holds $a \in I(s)$ iff $s \xrightarrow{a}$. $I(s)$ is called the *menu* of s . Intuitively, for $s \in S_n$, $I(s)$ is the set of actions that the process s can perform initially. Next, we define the result of testing a process with a given test. The informal explanation follows afterwards.

Definition 2. The function $\text{Res}: S \times T \mapsto \mathcal{R}$ that gives the result of testing a process s with a test T is defined as follows:

$$\text{Res}(s, T) = \begin{cases} 1, & \text{if } T \xrightarrow{\omega}, \\ \sum_{i \in I} \pi_i \cdot \text{Res}(s_i, T), & \text{if } s \xrightarrow{\pi_i} s_i \text{ for } i \in I \text{ and } T \not\xrightarrow{\omega} \\ \sum_{i \in I} \pi_i \cdot \text{Res}(s, T_i), & \text{if } T \xrightarrow{\pi_i} T_i \text{ for } i \in I \text{ and } s \not\xrightarrow{\omega} \\ \sum_{a \in K} \frac{a}{\sum_{b \in K} b} \cdot \text{Res}(s_a, T_a), & \text{for } K = I(s) \cap I(T), \text{ otherwise.} \end{cases}$$

As usual, the result of testing a process with a test denoting success is one, while the result of testing a process with a probabilistic state as a root (i.e. initially probabilistic process) is a weighted sum of the results of testing the subsequent processes with the same test. Similarly when the test is initially probabilistic. The novelty is in the result of testing an initially nondeterministic process s with a test T that can initially perform actions from \mathcal{A} only. Namely, when the process and the test synchronize on an action, the resulting transition is labeled with a “weighting factor”, containing information about the way this synchronization happened. This information has form of a rational function, the numerator of which represents the synchronized action itself, while the denominator is the sum of the common initial actions of s and T , i.e., all actions on which s and T could have synchronized at the current step. Then, the rational function is temporarily treated as “symbolic” probability, in order to compute the final result of the testing. The final result is again a rational function in \mathcal{R} .

Fig. 2 represents graphically the result of testing process s in Fig. 1 with the test u from the same figure. It is easy to compute that the result of testing is equal to $\frac{1}{2}$, which establishes one of our goals set in Section 1. However, in many cases the result is a non-scalar rational function. For example, denote by “+” the external choice operator. The result of applying test $h.p.\omega + t.\omega$ to each of processes s and \bar{s} in Fig. 1 is $\frac{h+2t}{2(h+t)}$.

Definition 3. Two processes s and \bar{s} are testing equivalent, notation $s \approx_{\mathcal{T}} \bar{s}$, iff $\text{Res}(s, T)$ and $\text{Res}(\bar{s}, T)$ are equal functions for every test T .

Obviously, comparing two results boils down to comparing two polynomials, after both rational functions have been transformed to equal denominators.

Example 1. Consider the processes in Fig. 3. The test $a.\omega + b.c.\omega$ distinguishes between the two processes.

Remark 1. Def. 2 assumes that, when the process and the test are ready to synchronize on an action, the test can see which actions have been offered from

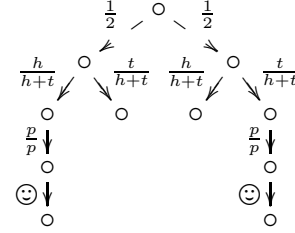


Fig. 2: Graphical representation of the result of testing s (Fig. 1) with u

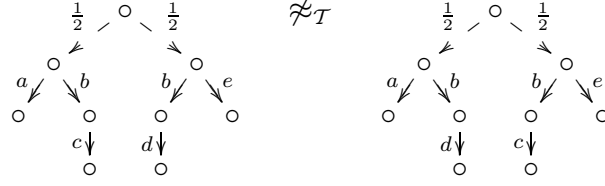


Fig. 3: Processes s (left) and \bar{s} (right) are not testing equivalent

the process. This corresponds to the user (e.g. u in Fig. 1) being able to see the menu that the machine (e.g. s in Fig. 1) offers. Note that this assumption does not exist in the standard non-probabilistic testing theory [6]. However, in real-life systems this is usually the case. Moreover, this assumption is mild with respect to probabilistic may/must testing approaches, where one needs to know the complete internal structure of the composed process, which, on the other side, yields unrealistic over-estimations of probabilities. In contrary, in our case, in order to compute the function $\text{Res}(s, T)$, it is not necessary that the probabilistic transitions of s and their labels are known. Their effect can be inferred statistically, by testing s with T sufficiently many times. To simplify the presentation, we do not go into details on statistical testing.

4 Probabilistic ready trace semantics

In this section we define a probabilistic version of ready trace equivalence [1, 21].

Definition 4 (Ready trace). A ready trace of length n is a sequence $\mathcal{O} = (M_1, a_1, M_2, a_2, \dots, M_{n-1}, a_{n-1}, M_n)$ where $M_i \in 2^{\mathcal{A}}$ for all $i \in \{1, 2, \dots, n\}$ and $a_i \in M_i$ for all $i \in \{1, 2, \dots, n-1\}$.

We assume that the observer has ability to observe the actions that the process performs, together with the menus out of which actions are chosen. Intuitively, a ready trace $\mathcal{O} = (M_1, a_1, M_2, a_2, \dots, M_{n-1}, a_{n-1}, M_n)$ can be observed if the initial menu is M_1 , then action $a_1 \in M_1$ is performed, then the next menu is M_2 , then action $a_2 \in M_2$ is performed and so on, until the observing ends at a point when the menu is M_n . It is essential that, since the probabilistic transitions are not observable, the observer cannot infer where exactly they happen in the ready trace.

Clearly the probability of observing a ready trace $(\{a, b\}, a, \{c\})$ is conditioned on choosing the action a from the menu $\{a, b\}$. This suggests that, when defining probabilities on ready traces, the Bayesian definition of probability is more appropriate than the measure-theoretic definition that is usually taken.

Next, given a process s , we define a process $s_{(M, a)}$. Intuitively, $s_{(M, a)}$ is the process that s becomes, assuming that menu M was offered to s and action a was performed.

Definition 5. Let s be a process graph. Let $M \subseteq \mathcal{A}$, $a \in M$ be such that $I(s) = M$ if $s \in S_n$ or otherwise there exists a transition $s \dashrightarrow s'$ such that $I(s') = M$. The process graph $s_{(M, a)}$ is obtained from s in the following way:

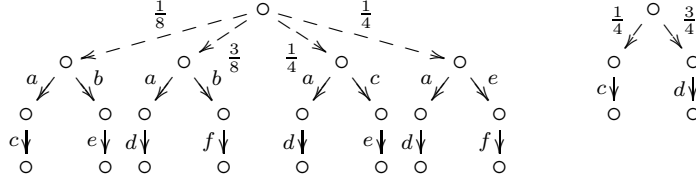


Fig. 4: Example of a process s (left) and $s_{(\{a,b\},a)}$ (right).

- if $s \in S_n$ then the root of $s_{(M,a)}$ is the state s' such that $s \xrightarrow{a} s'$, and
- if $s \in S_p$ then a new state $s_{(M,a)}$ is created. Let $\pi = \sum_{s \dashrightarrow s_i, I(s_i)=M} \pi_i$. For all s'_i such that $s \dashrightarrow s_i \xrightarrow{a} s'_i$ and $I(s_i) = M$:
 - if $s'_i \not\rightarrow$, then an edge $s_{(M,a)} \dashrightarrow s'_i$ is created;
 - for all transitions $s'_i \xrightarrow{\rho_i} s''_i$, an edge $s_{(M,a)} \dashrightarrow s''_i$ is created.

Example 2. Consider processes s and $s_{(\{a,b\},a)}$ in Fig. 4. Assuming that the initial menu of s was $\{a, b\}$ and action a was performed, process $s_{(\{a,b\},a)}$ describes the further behaviour of s : with probability $\frac{1}{8}/(\frac{1}{8} + \frac{3}{8}) = \frac{1}{4}$ action c is performed, while with probability $\frac{3}{8}/(\frac{1}{8} + \frac{3}{8}) = \frac{3}{4}$ action d is performed.

Definition 6. Let $(M_1, a_1, M_2, a_2, \dots, M_{n-1}, a_{n-1}, M_n)$ be a ready trace of length n and s be a process graph. Functions $P_s^1(M)$ and $P_s^n(M_n | M_1, a_1, \dots, M_{n-1}, a_{n-1})$ (for $n > 1$) are defined in the following way:

$$P_s^1(M) = \begin{cases} \sum_{s \dashrightarrow s'} \pi \cdot P_{s'}^1(M) & \text{if } s \in S_p, \\ 1 & \text{if } s \in S_n, I(s) = M, \\ 0 & \text{otherwise.} \end{cases}$$

$$P_s^2(M_2 | M_1, a_1) = \begin{cases} P_{s_{(M_1, a_1)}}^1(M_2) & \text{if } P_s^1(M_1) > 0, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

$$P_s^n(M_n | M_1, a_1, \dots, a_{n-1}) = \begin{cases} P_{s_{(M_1, a_1)}}^{n-1}(M_n | M_2, a_2, \dots, a_{n-1}) & \text{if } P_s^1(M_1) > 0, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Let the sample space consist of all possible menus and $s \in S$. Function $P_s^1(M)$ can be interpreted as the probability that the menu M is observed initially when process s starts executing. Let the sample space consist of all ready traces of length n and let $s \in S$. The function $P_s^n(M_n | M_1, a_1, \dots, M_{n-1}, a_{n-1})$ can be interpreted as the probability of the event $\{(M_1, a_1, \dots, M_{n-1}, a_{n-1}, M_n)\}$, given the event $\{(M_1, a_1, \dots, M_{n-1}, a_{n-1}, X) : X \in 2^A\}$, if observing ready traces of process s . It can be checked that these probabilities are well defined, i.e., they satisfy the axioms A1-A3 of Section 2.

Definition 7 (Probabilistic ready trace equivalence). *Two processes s and \bar{s} are probabilistically ready trace equivalent, notation $s \approx_{\mathcal{O}} \bar{s}$, iff:*

- for all M in $2^{\mathcal{A}}$, $P_s^1(M) = P_{\bar{s}}^1(M)$ and
- for all $n > 1$, $P_s^n(M_n | M_1, a_1, \dots, M_{n-1}, a_{n-1})$ is defined if and only if $P_{\bar{s}}^n(M_n | M_1, a_1, \dots, M_{n-1}, a_{n-1})$ is defined, and in that case $P_s^n(M_n | M_1, a_1, \dots, M_{n-1}, a_{n-1}) = P_{\bar{s}}^n(M_n | M_1, a_1, \dots, M_{n-1}, a_{n-1})$.

Informally, two processes s and \bar{s} are ready-trace equivalent iff for every n and every ready trace $(M_1, a_1, M_2, a_2, \dots, M_n)$, the probability to observe M_n , under condition that previously the sequence $(M_1, a_1, M_2, a_2, \dots, a_{n-1})$ was observed, is defined at the same time for both s and \bar{s} ; moreover, in case both probabilities are defined, they coincide. Note that it is straightforward to construct a black-box testing scenario [4, 10] for this ready-trace equivalence.

Example 3. Processes s and \bar{s} in Fig. 1 are ready-trace equivalent. Processes in Fig. 3 are not ready-trace equivalent: for process s it holds $P_s^2(\{c\} | \{a, b\}, b) = \frac{1}{2}$, while for process \bar{s} it holds $P_{\bar{s}}^2(\{c\} | \{a, b\}, b) = 0$.

5 Algebra

In this section we define an algebra CSP_p of finite processes using $\approx_{\mathcal{O}}$ as an underlying equivalence. The purpose is to show that $\approx_{\mathcal{O}}$ is congruence for the standard operators on the model of reactive probabilistic processes and that all operators distribute through probabilistic choice, as all operators distribute through internal choice in standard CSP [11]. As discussed in Sec. 1, we do not use hiding operator. For more discussions on including internal nondeterminism in general, please see Sec. 7.

The set of CSP_p processes \mathbf{P} is generated by the following grammar:

$$\mathbf{P} ::= \delta \mid \sum_{i \in I} a_i. \mathbf{P}_i \mid \bigoplus_{i \in I} \pi_i \mathbf{P}_i \mid \Theta \mathbf{P} \mid \mathbf{P} \parallel \mathbf{P} \mid \mathbf{P} \parallel_L \mathbf{P}$$

where $\delta \notin \mathcal{A}$ is a new symbol, $\{a_i\}_{i \in I} \subseteq \mathcal{A}$, $a_i \neq a_j$ for $i, j \in I$, $i \neq j$, $\pi_i \in (0, 1]$, $\sum_{i \in I} \pi_i = 1$, and $L \subseteq \mathcal{A}$ is the set of actions that appear both in the left and in the right process of the expression $\mathbf{P} \parallel_L \mathbf{P}$.

Let p, q, r, \dots range over CSP_p processes. The constant δ stands for the *empty* process. The process $a.p$ performs the action a and continues as process p (we write a rather than $a.\delta$). The *external choice* $\sum_{i \in I} a_i.p_i$ stands for a choice among the actions $\{a_i\}_{i \in I}$ and proceeds as process p_j if action a_j is chosen and executed. The *probabilistic choice* $\bigoplus_{i \in I} \pi_i p_i$ behaves as p_i with probability π_i for $i \in I$. The *priority* operator Θ assumes a partial order $>$ on \mathcal{A} . For actions a and b , we say a has higher priority than b iff $a > b$. Θ forces the process to always perform the action with the highest priority in the current menu. In a *synchronized parallel composition* $p \parallel q$, the processes operate in a lock-step

$\frac{}{\sum_{i \in I} a_i.p_i \xrightarrow{a_i} p_i}$	$\frac{p \xrightarrow{a} p', q \xrightarrow{a} q'}{p \parallel q \xrightarrow{a} p' \parallel q'}$	$\frac{p \xrightarrow{\pi} p', q \xrightarrow{\rho} q'}{p \parallel q \xrightarrow{\pi\rho} p' \parallel q'}$
$\frac{p \xrightarrow{\pi} p', q \not\xrightarrow{\pi}}{p \parallel q \xrightarrow{\pi} p' \parallel q}$	$\frac{p_k \not\xrightarrow{\pi}, k \in I}{\bigoplus_{i \in I} \pi_i p_i \xrightarrow{\pi_k} p_k}$	$\frac{p_k \xrightarrow{\rho_k} p'_k, k \in I}{\bigoplus_{i \in I} \pi_i p_i \xrightarrow{\pi_k \rho_k} p'_k}$
$\frac{a \notin L, p \xrightarrow{a} p', q \not\xrightarrow{a}}{p \parallel_L q \xrightarrow{a} p' \parallel_L q}$	$\frac{a \in L, p \xrightarrow{a} p', q \xrightarrow{a} q'}{p \parallel_L q \xrightarrow{a} p' \parallel_L q'}$	$\frac{p \xrightarrow{\pi} p', q \not\xrightarrow{\pi}}{p \parallel_L q \xrightarrow{\pi} p' \parallel_L q}$
$\frac{p \xrightarrow{\pi} p', q \xrightarrow{\rho} q'}{p \parallel_L q \xrightarrow{\pi\rho} p' \parallel_L q'}$	$\frac{p \xrightarrow{a} p', p \not\xrightarrow{b} \text{ for } a < b}{\Theta p \xrightarrow{a} \Theta p'}$	$\frac{p \xrightarrow{\pi} p'}{\Theta p \xrightarrow{\pi} \Theta p'}$

Table 1: Operational semantics for CSP_p processes

synchronization. In a *parallel composition* $p \parallel_L q$, the processes synchronize on their common actions, while the other actions are interleaved.^{3 4}

Table 1 represents the operational semantics of CSP_p processes (we omit the symmetric rules for \parallel_L and \parallel).

As usual, a context is a CSP_p process with a hole in it. Given a context $C[\cdot]$ and a process p , we write $C[p]$ to denote the process obtained by filling in the hole of $C[\cdot]$ with p .

Theorem 1 (Congruence). *The equivalence $\approx_{\mathcal{O}}$ is congruence for the operators of CSP_p , i.e., if $p \approx_{\mathcal{O}} \bar{p}$ then for each context $C[\cdot]$, it holds that $C[p] \approx_{\mathcal{O}} C[\bar{p}]$.*

Proof. We prove the congruence result for parallel composition, because this is the most complicated case. We prove that if $p \approx_{\mathcal{O}} \bar{p}$ then $p \parallel_L q \approx_{\mathcal{O}} \bar{p} \parallel_L q$. Denote by L the set of the common actions for p and q (and therefore \bar{p} and q). Without loss of generality, assume that p, \bar{p} , and q are probabilistic processes. For arbitrary menus M', M'' , denote by $M' \otimes M''$ the menu $(M' \cup M'') \setminus (L \setminus (M' \cap M''))$.

By induction on n , we prove that if $p \approx_{\mathcal{O}} \bar{p}$ then $P_{(p \parallel_L q)}^n(M_n | M_1, a_1, \dots, M_{n-1}, a_{n-1}) = P_{(\bar{p} \parallel_L q)}^n(M_n | M_1, a_1, \dots, M_{n-1}, a_{n-1})$.

For arbitrary menus M_p and M_q , we have $P_p^1(M_p) = P_{\bar{p}}^1(M_p)$. Let M be a menu such that $P_{p \parallel_L q}^1(M) > 0$. This means that there exist menus M_p, M_q such

³ To preserve associativity of \parallel_L , we require that for any processes p, q , and r , if p and q share actions and q and r share actions then p and r do not share actions.

⁴ Sequential composition and successful termination can be also defined, which we avoid here to shorten.

that $P_p^1(M_p) > 0$, $P_q^1(M_q) > 0$, and $M = M_p \otimes M_q$ (by Table 1). We have,

$$\begin{aligned}
P_{p \parallel_L q}^1(M) &= \sum_{\substack{p \parallel_L q \xrightarrow{\lambda_k} r_k, \\ I(r_k) = M}} \lambda_k = \sum_{\substack{p \xrightarrow{\pi_i} p_i, q \xrightarrow{\rho_j} q_j, \\ I(p_i) \otimes I(q_j) = M}} \pi_i \cdot \rho_j \\
&= \sum_{q \xrightarrow{\rho_j} q_j} \rho_j \sum_{p \xrightarrow{\pi_i} p_i, M=I(p_i) \otimes I(q_j)} \pi_i = \sum_{q \xrightarrow{\rho_j} q_j} \rho_j \sum_{\bar{p} \xrightarrow{\bar{\pi}_i} \bar{p}_i, M=I(\bar{p}_i) \otimes I(q_j)} \bar{\pi}_i = P_{\bar{p} \parallel_L q}^1(M).
\end{aligned}$$

Suppose $P_{(p \parallel_L q)}^k(M_k | M_1, a_1, \dots, M_{k-1}, a_{k-1}) = P_{(\bar{p} \parallel_L q)}^k(M_k | M_1, a_1, \dots, M_{k-1}, a_{k-1})$ if $p \approx_{\mathcal{O}} \bar{p}$ and $k < n$.

Case 1 Suppose first that both $P_{(p \parallel_L q)}^n(M_n | M_1, a_1, \dots, M_{n-1}, a_{n-1})$ and $P_{(\bar{p} \parallel_L q)}^n(M_n | M_1, a_1, \dots, M_{n-1}, a_{n-1})$ are defined. Because of Def. 6 and the inductive assumption, it is enough to prove that $P_{(p \parallel_L q)(M_1, a_1)}^{n-1}(M_n | M_2, a_2, \dots, M_{n-1}, a_{n-1}) = P_{(\bar{p} \parallel_L q)(M_1, a_1)}^{n-1}(M_n | M_2, a_2, \dots, M_{n-1}, a_{n-1})$. Because of the inductive assumption, to prove the last, it is enough to prove that $(p \parallel_L q)_{(M_1, a_1)} \approx_{\mathcal{O}} (\bar{p} \parallel_L q)_{(M_1, a_1)}$.

Case 1.1 $a_1 = a \in L$.

Denote $\sum_{p \xrightarrow{\pi_i} p_i, q \xrightarrow{\rho_j} q_j, I(p_i) \otimes I(q_j) = M_1} \pi_i \rho_j$ by α . By Def. 5 and the rules in Table 1, we have

$$(p \parallel_L q)_{(M_1, a)} \equiv \bigoplus_{\substack{p \xrightarrow{\pi_i} p_i, q \xrightarrow{\rho_j} q_j, \\ I(p_i) \otimes I(q_j) = M_1}} \frac{\pi_i \rho_j}{\alpha} \left(p_{i(I(p_i), a)} \parallel_L q_{j(I(q_j), a)} \right). \quad (1)$$

On the other hand, denoting $\sum_{M_1=M_p \otimes M_q} P_p(M_p)P_q(M_q)$ by β , we have

$$\begin{aligned}
& \bigoplus_{M_1=M_p \otimes M_q} \frac{P_p(M_p)P_q(M_q)}{\beta} \left(p_{(M_p,a)} \parallel_L q_{(M_q,a)} \right) \\
& \equiv \left(\bigoplus_{M_1=M_p \otimes M_q} \frac{P_p(M_p)P_q(M_q)}{\beta} \right) \times \\
& \quad \times \left(\bigoplus_{\substack{p \dashrightarrow p_i, I(p_i)=M_p \\ q \dashrightarrow q_j, I(q_j)=M_q}} \frac{\pi_i}{P_p(M_p)} p_{i(M_p,a)} \right) \parallel_L \left(\bigoplus_{\substack{q \dashrightarrow q_j, I(q_j)=M_q}} \frac{\rho_j}{P_q(M_q)} q_{j(M_q,a)} \right) \\
& \equiv \left(\bigoplus_{M_1=M_p \otimes M_q} \frac{P_p(M_p)P_q(M_q)}{\beta} \right) \times \\
& \quad \times \left(\bigoplus_{\substack{p \dashrightarrow p_i, I(p_i)=M_p, \\ q \dashrightarrow q_j, I(q_j)=M_q}} \frac{\pi_i \rho_j}{P_p(M_p)P_q(M_q)} \left(p_{i(M_p,a)} \parallel_L q_{j(M_q,a)} \right) \right) \\
& \equiv \bigoplus_{\substack{p \dashrightarrow p_j, q \dashrightarrow q_j, \\ M_1 = I(p_i) \otimes I(q_j)}} \frac{\pi_i \rho_j}{\alpha} \left(p_{i(I(p_i),a)} \parallel_L q_{j(I(q_j),a)} \right). \tag{2}
\end{aligned}$$

From (1) and (2) we have

$$(p \parallel_L q)_{(M_1,a)} \equiv \bigoplus_{M_p, M_q: M_1=M_p \otimes M_q} \frac{P_p(M_p)P_q(M_q)}{\sum_{M_p, M_q} P_p(M_p)P_q(M_q)} \left(p_{(M_p,a)} \parallel_L q_{(M_q,a)} \right). \tag{3}$$

Similarly,

$$(\bar{p} \parallel_L q)_{(M_1,a)} \equiv \bigoplus_{M_p, M_q: M_1=M_p \otimes M_q} \frac{P_{\bar{p}}(M_p)P_q(M_q)}{\sum_{M_p, M_q} P_{\bar{p}}(M_p)P_q(M_q)} \left(\bar{p}_{(M_p,a)} \parallel_L q_{(M_q,a)} \right). \tag{4}$$

From the inductive assumption and because $p \approx_{\mathcal{O}} \bar{p}$ and $\approx_{\mathcal{O}}$ is congruence for \oplus , we have

$$\begin{aligned}
& \bigoplus_{M_p, M_q: M_1=M_p \otimes M_q} \frac{P_p(M_p)P_q(M_q)}{\sum_{M_p, M_q} P_p(M_p)P_q(M_q)} \left(p_{(M_p,a)} \parallel_L q_{(M_q,a)} \right) \\
& \equiv \bigoplus_{M_p, M_q: M_1=M_p \otimes M_q} \frac{P_{\bar{p}}(M_p)P_q(M_q)}{\sum_{M_p, M_q} P_{\bar{p}}(M_p)P_q(M_q)} \left(\bar{p}_{(M_p,a)} \parallel_L q_{(M_q,a)} \right). \tag{5}
\end{aligned}$$

From (3), (4), and (5) it follows that $(p \parallel_L q)_{(M_1,a)} \equiv (\bar{p} \parallel_L q)_{(M_1,a)}$.

Case 1.2 $a_1 \notin L$, a_1 appears in p . The proof is similar to Case 1, with the difference that instead of a process $q_{(M_q, a_1)}$, we use a process $q_{(M_q)}$. The last one is defined by a process graph obtained in a similar way as $q_{(M_q, a_1)}$, with the exception that $q_{(M_q)}$ is “ready” to choose any action from the menu M_q .

Case 1.3 $a_1 \notin L$, a_1 appears in q - symmetric to Case 2.

Case 2 Suppose now that $P_{(p \parallel_L q)}^k(M_k | M_1, a_1, \dots, M_{k-1}, a_{k-1})$ is defined but $P_{(\bar{p} \parallel_L q)}^k(M_k | M_1, a_1, \dots, M_{k-1}, a_{k-1})$ is not defined. Either $P_{(p \parallel_L q)}(M_1) > 0$ while $P_{(\bar{p} \parallel_L q)}(M_1) = 0$, which is not possible because $p \approx_{\mathcal{O}} \bar{p}$, or $P_{(p \parallel_L q)(M_1, a_1)}^{k-1}(M_k | M_2, a_2, \dots, M_{k-1}, a_{k-1})$ is defined but $P_{(\bar{p} \parallel_L q)(M_1, a_1)}^{k-1}(M_k | M_2, a_2, \dots, M_{k-1}, a_{k-1})$ is not defined, which again is not possible because of the inductive assumption.

The following two theorems formulate the laws of distributivity of the operators over probabilistic choice.

Theorem 2. For processes $\{x_{ij}\}_{i \in I, j \in J}$ and actions $\{a_i\}_{i \in I} \subseteq \mathcal{A}$, it holds $\sum_{i \in I} a_i \cdot \bigoplus_{j \in J} \pi_j x_{ij} \approx_{\mathcal{O}} \bigoplus_{j \in J} \pi_j \sum_{i \in I} a_i \cdot x_{ij}$.

Proof. Let $M = \{a_i\}_{i \in I}$, $p \equiv \sum_{i \in I} a_i \cdot \bigoplus_{j \in J} \pi_j x_{ij}$ and $\bar{p} \equiv \bigoplus_{j \in J} \pi_j \sum_{i \in I} a_i \cdot x_{ij}$. Then, it is easy to show that, for every $i \in I$, $p_{(M, a_i)} \approx_{\mathcal{O}} \bar{p}_{(M, a_i)}$. Let $n > 1$ and (M_1, b_1, \dots, M_n) be an observation. Then,

$$P_p^n(M_n | M_1, b_1, \dots, b_{n-1}) = \begin{cases} P_{p_{(M_1, b_1)}}^{n-1}(M_n | M_2, b_2, \dots, b_{n-1}) & \text{if } M_1 = M, b_1 \in M \\ \text{undefined} & \text{otherwise,} \end{cases}$$

and

$$P_{\bar{p}}^n(M_n | M_1, b_1, \dots, b_{n-1}) = \begin{cases} P_{\bar{p}_{(M_1, b_1)}}^{n-1}(M_n | M_2, b_2, \dots, b_{n-1}) & \text{if } M_1 = M, b_1 \in M \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Now, it easily follows that $p \approx_{\mathcal{O}} \bar{p}$.

Theorem 3. For every context $C[\cdot]$, it holds $C[\bigoplus_{i \in I} \pi_i x_i] \approx_{\mathcal{O}} \bigoplus_{i \in I} \pi_i C[x_i]$.

Proof. By structural induction, similarly to the proof of Theorem 2.

6 Relationship between $\approx_{\mathcal{T}}$ and $\approx_{\mathcal{O}}$

We establish our main result, namely that the testing equivalence $\approx_{\mathcal{T}}$ coincides with the probabilistic ready trace equivalence $\approx_{\mathcal{O}}$. As an intermediate result, we prove that probabilistic transitions do not add distinguishing power to the tests.

Theorem 4. Let s and t be two processes. If $s \approx_{\mathcal{O}} t$ then $s \approx_{\mathcal{T}} t$.

Proof. Suppose $s \not\approx_{\mathcal{T}} t$. There exists a test T such that $\text{Res}(s, T) \neq \text{Res}(t, T)$. W.l.g., assume that s and t start with probabilistic transitions. By Def. 2,

$$\text{Res}(s, T) = \sum_{T \xrightarrow{\rho_j} T_j} \rho_j \sum_{s \xrightarrow{\pi_i} s_i} \pi_i \sum_{a \in I(s_i) \cap I(T_j)} \frac{a}{\sum_{b \in I(s_i) \cap I(T_j)} b} \cdot \text{Res}(s_{ia}, T_{ja}). \quad (6)$$

By Def. 5, from (6) we obtain

$$\begin{aligned} \text{Res}(s, T) = & \sum_{M': P_T^1(M') > 0} P_T^1(M') \sum_{M: P_s^1(M) > 0} P_s^1(M) \times \\ & \times \sum_{a \in M \cap M'} \frac{a}{\sum_{b \in M \cap M'} b} \text{Res}(s_{(M,a)}, T_{(M',a)}). \end{aligned} \quad (7)$$

Similarly we obtain

$$\begin{aligned} \text{Res}(t, T) = & \sum_{M': P_T^1(M') > 0} P_T^1(M') \sum_{M: P_t^1(M) > 0} P_t^1(M) \times \\ & \times \sum_{a \in M \cap M'} \frac{a}{\sum_{b \in M \cap M'} b} \text{Res}(t_{(M,a)}, T_{(M',a)}). \end{aligned} \quad (8)$$

Now, assume $s \approx_{\mathcal{O}} t$. Define a *length of a test* to be the length of the longest sequence of actions the test can perform before executing the action ω . The proof is by induction on the minimal length of a nonprobabilistic test that distinguishes between s and t .

Let T be a test of length 1 such that $\text{Res}(s, T) \neq \text{Res}(t, T)$. From Def. 2 it follows that for every process u ,

$$\text{Res}(u, T_{(M',a)}) = P_{T_{(M',a)}}^1(\{\omega\}). \quad (9)$$

From (7) and (9) we have

$$\begin{aligned} \text{Res}(s, T) = & \sum_{M': P_T^1(M') > 0} P_T^1(M') \sum_{M: P_s^1(M) > 0} P_s^1(M) \times \\ & \times \sum_{a \in M \cap M'} \frac{a}{\sum_{b \in M \cap M'} b} P_{T_{(M',a)}}^1(\{\omega\}). \end{aligned} \quad (10)$$

Similarly we obtain

$$\begin{aligned} \text{Res}(t, T) = & \sum_{M': P_T^1(M') > 0} P_T^1(M') \sum_{M: P_t^1(M) > 0} P_t^1(M) \times \\ & \times \sum_{a \in M \cap M'} \frac{a}{\sum_{b \in M \cap M'} b} P_{T_{(M',a)}}^1(\{\omega\}). \end{aligned} \quad (11)$$

From (10),(11) and from the assumption that $P_s^1(M) = P_t^1(M)$ for every menu M , we obtain that $\text{Res}(s, T) = \text{Res}(t, T)$, i.e. we obtain contradiction. Therefore, there exists a menu M such that $P_s^1(M) \neq P_t^1(M)$, i.e. $s \not\sim_{\mathcal{O}} t$.

Let T be a test of length greater than one such that $\text{Res}(s, T) \neq \text{Res}(t, T)$. If there exists a menu M such that $P_s^1(M) \neq P_t^1(M)$, then $s \not\sim_{\mathcal{O}} t$ and the proof is over. Therefore, suppose $P_s^1(M) = P_t^1(M)$ for every menu $M \subseteq \mathcal{A}$. From (7) and (8) we have that for some menus M, M' and action $a \in M \cap M'$, it holds $\text{Res}(s_{(M,a)}, T_{(M',a)}) \neq \text{Res}(t_{(M,a)}, T_{(M',a)})$. Now, by the inductive assumption, we have $s_{(M,a)} \not\sim_{\mathcal{O}} t_{(M,a)}$, i.e. there exists a ready trace (M_2, a_2, \dots, M_k) such that $P_{s_{(M,a)}}^{k-1}(M_k | M_2, a_2, \dots, a_{k-1}) \neq P_{t_{(M,a)}}^{k-1}(M_k | M_2, a_2, \dots, a_{k-1})$ (or they are not defined at the same time). From the last, from the assumption that $P_s^1(M) = P_t^1(M) > 0$, and from Def. 6 it follows that $P_s^k(M_k | M, a, M_2, a_2, \dots, a_{k-1}) \neq P_t^k(M_k | M, a, M_2, a_2, \dots, a_{k-1})$ (or they are not defined at the same time), i.e. $s \not\sim_{\mathcal{O}} t$. This completes the proof of the theorem.

The following lemma, which considers the determinant of a certain type of an almost-triangular matrix, shall be needed in the proof of Theorem 5.

Lemma 1. *Let \mathbf{Q} be a square $n \times n$ matrix with elements q_{ij} , for $1 \leq i \leq n$ and $1 \leq j \leq n$. Suppose $q_{ij} \in \{0, 1\}$ for $i > 1$, $q_{ij} = 1$ for $i = j + 1$, $q_{ij} = 0$ for $i > j + 1$, and $q_{1j} = \frac{Q_1}{Q_j}$ for $1 \leq j \leq n$, where $Q_1, Q_2 \dots Q_n$ are irreducible, mutually prime polynomials with positive variables, and of non-zero degrees. Then the determinant of \mathbf{Q} is a non-zero rational function.*

Proof. The determinant $\text{Det}(\mathbf{Q})$ of matrix \mathbf{Q} can be obtained from the general recursive formula $\text{Det}(\mathbf{Q}) = \sum_{j=1}^n (-1)^{1+j} q_{1j} \text{Det}(\mathbf{Q}_{1j})$, where \mathbf{Q}_{1j} is the matrix obtained by deleting the first row and the j -th column of \mathbf{Q} . Observe that \mathbf{Q}_{1n} is an upper-triangular matrix, the diagonal elements of which are all equal to one. Since the determinant of a triangular matrix is equal to the product of its diagonal elements, we have $\text{Det}(\mathbf{Q}_{1n}) = 1$. Therefore, the coefficient in front of the rational function $\frac{Q_1}{Q_n}$ in $\text{Det}(\mathbf{Q})$ is equal to 1. Suppose $\text{Det}(\mathbf{Q})$ is a zero-function. Then, the rational function $\frac{1}{Q_n}$ is equal to a linear combination of $\frac{1}{Q_1}, \dots, \frac{1}{Q_{n-1}}$. This means that the rational function $\frac{Q_1 \cdot Q_2 \cdot \dots \cdot Q_{n-1}}{Q_n}$ is a polynomial. The last is impossible, since, by assumption, the denominator is irreducible polynomial of non-zero degree and is not contained in the numerator. Therefore, $\text{Det}(\mathbf{Q})$ is not a zero-function.

Theorem 5. *Let s and t be two processes such that $s \not\sim_{\mathcal{O}} t$. There exists a test T that has no probabilistic transitions such that $\text{Res}(s, T) \neq \text{Res}(t, T)$.*

Proof. We prove the theorem by induction on the minimal length m of a ready trace that distinguishes between s and t . For $m = 1$, we prove that the test $T = \sum_{a \notin M} a.\omega$, where M is a menu with a minimal possible number of actions such that $P_s^1(M) \neq P_t^1(M)$, distinguishes between s and t . For $m > 1$ the proof goes as follows. If $P_s^1(M) = P_t^1(M)$ for every menu M , then by the inductive assumption

it follows that there exists a test T_1 , menu M_1 and action $a_1 \in M_1$ such that $\text{Res}(s_{(M_1, a_1)}, T_1) \neq \text{Res}(t_{(M_1, a_1)}, T_1)$. We show that there exists a subset of the action set, say Act , such that the test $T = a_1.T_1 + \sum_{b \in \text{Act}} \omega$ distinguishes between s and t . To prove this, we take M_1 to be the menu containing a minimal possible number of actions such that $P_s^1(M_1) > 0$, $a_1 \in M_1$, and $\text{Res}(s_{(M_1, a_1)}, T_1) \neq \text{Res}(t_{(M_1, a_1)}, T_1)$. Then we take the set Act' to consist of the actions that can be initially performed by s but do not belong to menu M_1 . Then, we show that there must exist a subset Act of Act' such that the test $T = a_1.T_1 + \sum_{b \in \text{Act}} \omega$ distinguishes between s and t (otherwise, we obtain that $\text{Res}(s_{(M_1, a_1)}, T_1) = \text{Res}(t_{(M_1, a_1)}, T_1)$, which contradicts our assumption).

We now proceed with a detailed presentation of the proof.

From $s \not\approx_{\mathcal{O}} t$ and by Def. 7, there must exist a ready trace (M_1, a_1, \dots, M_m) such that $P_s^m(M_m | M_1, a_1, \dots, a_{m-1}) \neq P_t^m(M_m | M_1, a_1, \dots, a_{m-1})$. The proof is by induction on m .

Case 1 ($m = 1$) Suppose first that there exists a menu M such that $P_s^1(M) \neq P_t^1(M)$. Let M be a menu with a minimal possible number of actions such that $P_s^1(M) \neq P_t^1(M)$. Take $T = \sum_{a \notin M} a.\omega$. We have $\text{Res}(s, T) = 1 - \sum_{M' \subseteq M} P_s^1(M')$, because the actions of s and T will fail to synchronize if and only if the random choice decides that menu M or some menu $M' \subset M$ is offered to process s initially. Similarly, $\text{Res}(t, T) = 1 - \sum_{M' \subseteq M} P_t^1(M')$. Now, suppose that $\text{Res}(s, T) = \text{Res}(t, T)$. We have $\sum_{M' \subseteq M} P_s^1(M') = \sum_{M' \subseteq M} P_t^1(M')$. From this and $P_s^1(M) \neq P_t^1(M)$, it follows that there exists a menu $M' \subset M$ such that also $P_s^1(M') \neq P_t^1(M')$. But this contradicts the assumption that M is a menu with a minimal possible number of actions such that $P_s^1(M) \neq P_t^1(M)$.

Case 2 ($m > 1$) Suppose now that $P_s^1(M) = P_t^1(M)$ for every menu M . Let (M_1, a_1, \dots, M_m) be a ready trace such that $P_s^{m-1}(M_m | M_1, a_1, \dots, a_{m-1}) \neq P_t^{m-1}(M_m | M_1, a_1, \dots, a_{m-1})$. From $P_s^1(M_1) = P_t^1(M_1)$, and from Definitions 5 and 6, it follows that $P_{s_{(M_1, a_1)}}^{m-1}(M_m | M_2, a_2, \dots, a_{m-1}) \neq P_{t_{(M_1, a_1)}}^{m-1}(M_m | M_2, a_2, \dots, a_{m-1})$ (in case $m = 2$, $P_{s_{(M_1, a_1)}}^1(M_2) \neq P_{t_{(M_1, a_1)}}^1(M_2)$). Now, by the inductive assumption, there exists a non-probabilistic test T_1 such that $\text{Res}(s_{(M_1, a_1)}, T_1) \neq \text{Res}(t_{(M_1, a_1)}, T_1)$.

Case 2.1 Suppose first that a_1 does not belong to any first-level menu of s other than M_1 , i.e. that for every menu M , $P_s^1(M) > 0$ and $a_1 \in M$ implies $M = M_1$. Then the test $T = a_1.T_1$ distinguishes between s and t .

Case 2.2 Suppose now that a_1 belongs to at least one first-level menu of s other than M_1 , i.e. there exists at least one menu $M \neq M_1$ such that $P_s^1(M) > 0$ and $a_1 \in M$. Without loss of generality, assume that M_1 is a menu with a minimal possible number of actions such that $P_s^1(M_1) > 0$, $a_1 \in M_1$, and $\text{Res}(s_{(M_1, a_1)}, T_1) \neq \text{Res}(t_{(M_1, a_1)}, T_1)$. Let $\{b_j\}_{j \in J}$ be the set of actions that appear in the first level of s (and therefore t) but not in M_1 , i.e. $b \in \{b_j\}_{j \in J}$ if

and only if $b \notin M_1$ and there exists a menu M such that $P_s^1(M) > 0$, $b \in M$. We shall prove that there exists $J' \subseteq J$ such that the test $T = a_1.T_1 + \sum_{j \in J'} b_j.\omega$ distinguishes between s and t . More concretely, we shall prove that, assuming the opposite, it follows that $\text{Res}(s_{(M_1, a_1)}, T_1) = \text{Res}(t_{(M_1, a_1)}, T_1)$, thus obtaining contradiction.

Case 2.2.a Suppose first that $\{b_j\}_{j \in J} = \emptyset$. This means that there are no actions other than those in M_1 , that appear in the first level of s . Therefore, all menus M for which $P_s^1(M) > 0$ satisfy $M \subseteq M_1$. We prove that the test $T = a_1.T_1$ distinguishes between s and t . Assume that $\text{Res}(s, T) = \text{Res}(t, T)$. From the last and from Def. 2, we obtain

$$\sum_{M: P_s^1(M) > 0, a_1 \in M \subseteq M_1} (\text{Res}(s_{(M, a_1)}, T_1) - \text{Res}(t_{(M, a_1)}, T_1)) = 0. \quad (12)$$

By assumption, for every $M \subset M_1$ such that $a_1 \in M$ it holds $\text{Res}(s_{(M, a_1)}, T_1) = \text{Res}(t_{(M, a_1)}, T_1)$. Therefore, from (12) we obtain $\text{Res}(s_{(M_1, a_1)}, T_1) = \text{Res}(t_{(M_1, a_1)}, T_1)$, which contradicts the assumption $\text{Res}(s_{(M_1, a_1)}, T_1) \neq \text{Res}(t_{(M_1, a_1)}, T_1)$.

Case 2.2.b Suppose now that $\{b_j\}_{j \in J} \neq \emptyset$. Given action $b_i \in \{b_j\}_{j \in J}$, denote by \mathcal{M}_i the set of all first-level menus of s that contain b_i and a_1 , i.e. $M \in \mathcal{M}_i$ iff $P_s^1(M) > 0$ and $b_i, a_1 \in M$; denote by \mathcal{M}_i^C the set of all first-level menus of s that do not contain b_i but have a_1 , i.e. $M \in \mathcal{M}_i^C$ iff $P_s^1(M) > 0$, $b_i \notin M$ and $a_1 \in M$.

Let $T = a_1.T_1 + \sum_{j \in J'} b_j.\omega$ for some $J' = \{1, 2, \dots, n\} \subseteq J$ and suppose $\text{Res}(s, T) = \text{Res}(t, T)$. Since $P_s^1(M) = P_t^1(M)$ for every menu M , observe that only if action a_1 is performed initially, it is possible for the test $T = a_1.T_1 + \sum_{j \in J'} b_j.\omega$ to make a difference between s and t . Because of this and by Definitions 2 and 5 it follows that

$$\begin{aligned} & \sum_{M \in \mathcal{M}_n^C \cap \mathcal{M}_{n-1}^C \cap \dots \cap \mathcal{M}_1^C} \frac{a_1}{a_1} P_s^1(M) (\text{Res}(s_{(M, a_1)}, T_1) - \text{Res}(t_{(M, a_1)}, T_1)) \\ & + \sum_{M \in \mathcal{M}_n^C \cap \mathcal{M}_{n-1}^C \cap \dots \cap \mathcal{M}_1} \frac{a_1}{a_1 + b_1} P_s^1(M) (\text{Res}(s_{(M, a_1)}, T_1) - \text{Res}(t_{(M, a_1)}, T_1)) \\ & + \dots \\ & + \sum_{M \in \mathcal{M}_n \cap \dots \cap \mathcal{M}_1} \frac{a_1}{a_1 + \sum_{j=1}^n b_j} P_s^1(M) (\text{Res}(s_{(M, a_1)}, T_1) - \text{Res}(t_{(M, a_1)}, T_1)) \\ & = 0. \end{aligned} \quad (13)$$

Each intersection appearing under the \sum -operators of (13) can be mapped bijectively to a binary number of n digits – the i -th digit being 0 if the intersection contains \mathcal{M}_{n+1-i}^C , and 1 if the intersection contains \mathcal{M}_{n+1-i} . (For reasons that will become clear later, the order of the indexing is reversed.)

Suppose $\text{Res}(s, T) = \text{Res}(t, T)$ for every test $T = a_1.T_1 + \sum_{j \in J'} b_j.\omega$, where $J' \subseteq J$. We shall prove that, in this case, every sum $\sum (\text{Res}(s_{(M, a_1)}, T_1) - \text{Res}(t_{(M, a_1)}, T_1))$ that appears in (13) when $J' = J$ is equal to a zero-function. In particular, the equality

$$\sum_{M \in \bigcap_{j \in J} \mathcal{M}_j^C} (\text{Res}(s_{(M, a_1)}, T_1) - \text{Res}(t_{(M, a_1)}, T_1)) = 0 \quad (14)$$

will hold. Note that the set $\bigcap_{j \in J} \mathcal{M}_j^C$ contains all first-level menus of s that have the action a_1 but do not have any other action that does not appear in M_1 . Therefore, $\bigcap_{j \in J} \mathcal{M}_j^C$ consists of the subsets of M_1 that contain a_1 . Thus, the equation (14) is equivalent to the equation (12) which leads to $\text{Res}(s_{(M_1, a_1)}, T_1) = \text{Res}(t_{(M_1, a_1)}, T_1)$, i.e. to contradiction. This would complete the proof of the theorem.

We now proceed with proving the above stated claim. We prove a more general result, namely that for $J' \subseteq J$, under assumption that $\text{Res}(s, T) = \text{Res}(t, T)$ for every test $T = a_1.T_1 + \sum_{i \in J'} b_i.\omega$ such that $J'' \subseteq J$ and $|J''| \leq |J'|$, it holds that every sum $\sum (\text{Res}(s_{(M, a_1)}, T_1) - \text{Res}(t_{(M, a_1)}, T_1))$ that appears in (13) is equal to zero.

Suppose first that $|J'| = 1$, i.e. $J' = \{1\}$. Assume that

$$\text{Res}(s, a_1.T_1) = \text{Res}(t, a_1.T_1) \quad (15)$$

and

$$\text{Res}(s, a_1.T_1 + b_1.\omega) = \text{Res}(t, a_1.T_1 + b_1.\omega). \quad (16)$$

From (15), Def. 2, and because $P_s^1(M) = P_t^1(M)$ for every menu M , we obtain

$$\sum_{M \in \mathcal{M}_1 \cup \mathcal{M}_1^C} \frac{a_1}{a_1} P_s^1(M) (\text{Res}(s_{(M, a_1)}, T_1) - \text{Res}(t_{(M, a_1)}, T_1)) = 0. \quad (17)$$

The equation (13) turns into

$$\begin{aligned} & \sum_{M \in \mathcal{M}_1^C} \frac{a_1}{a_1} P_s^1(M) (\text{Res}(s_{(M, a_1)}, T_1) - \text{Res}(t_{(M, a_1)}, T_1)) \\ & + \sum_{M \in \mathcal{M}_1} \frac{a_1}{a_1 + b_1} P_s^1(M) (\text{Res}(s_{(M, a_1)}, T_1) - \text{Res}(t_{(M, a_1)}, T_1)) \\ & = 0. \end{aligned} \quad (18)$$

Denote $\sum_{M \in \mathcal{M}_1^C} P_a^1(M) (\text{Res}(s_{(M, a_1)}, T_1) - \text{Res}(t_{(M, a_1)}, T_1))$ by x_0 and $\sum_{M \in \mathcal{M}_1} P_a^1(M) (\text{Res}(s_{(M, a_1)}, T_1) - \text{Res}(t_{(M, a_1)}, T_1))$ by x_1 . Our goal is to show that $x_0 = 0$ and $x_1 = 0$, i.e. that they are zero-functions. From (17) and (18) we obtain the following system of equations for the unknowns x_0 and x_1 :

$$\begin{cases} \frac{a_1}{a_1} x_0 + \frac{a_1}{a_1 + b_1} x_1 = 0 \\ x_0 + x_1 = 0, \end{cases}$$

or in a matrix form

$$\mathbf{Q}_1 \mathbf{x} = \mathbf{0},$$

where

$$\mathbf{Q}_1 = \begin{pmatrix} \frac{a_1}{a_1} & \frac{a_1}{a_1+b_1} \\ 1 & 1 \end{pmatrix}, \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \text{ and } \mathbf{0} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Since the determinant of the matrix \mathbf{Q}_1 is not a zero-function, it follows that $x_0 = 0$ and $x_1 = 0$ is the only solution of the system.

To present a better intuition on the proof in the general case, we shall also consider separately the case $|J'| = 2$. Let $J' = \{1, 2\}$ and assume that $\text{Res}(s, T) = \text{Res}(t, T)$ for every test $T = a_1.T_1 + \sum_{i \in J''} b_i.\omega$ such that $J'' \subseteq J$ and $|J''| \leq |J'|$. The equation (13) turns into

$$\begin{aligned} & \sum_{M \in \mathcal{M}_2^C \cap \mathcal{M}_1^C} \frac{a_1}{a_1} P_s^1(M) (\text{Res}(s_{(M,a_1)}, T_1) - \text{Res}(t_{(M,a_1)}, T_1)) \\ & + \sum_{M \in \mathcal{M}_2^C \cap \mathcal{M}_1} \frac{a_1}{a_1 + b_1} P_s^1(M) (\text{Res}(s_{(M,a_1)}, T_1) - \text{Res}(t_{(M,a_1)}, T_1)) \\ & + \sum_{M \in \mathcal{M}_2 \cap \mathcal{M}_1^C} \frac{a_1}{a_1 + b_2} P_s^1(M) (\text{Res}(s_{(M,a_1)}, T_1) - \text{Res}(t_{(M,a_1)}, T_1)) \\ & + \sum_{M \in \mathcal{M}_2 \cap \mathcal{M}_1} \frac{a_1}{a_1 + b_1 + b_2} P_s^1(M) (\text{Res}(s_{(M,a_1)}, T_1) - \text{Res}(t_{(M,a_1)}, T_1)) \\ & = 0. \end{aligned} \tag{19}$$

Denoting $\sum_{M \in \mathcal{M}_2^C \cap \mathcal{M}_1^C} P_s^1(M) (\text{Res}(s_{(M,a_1)}, T_1) - \text{Res}(t_{(M,a_1)}, T_1))$ by x_{00} and so on, (19) turns into

$$\frac{a_1}{a_1} x_{00} + \frac{a_1}{a_1 + b_1} x_{01} + \frac{a_1}{a_1 + b_2} x_{10} + \frac{a_1}{a_1 + b_1 + b_2} x_{11} = 0. \tag{20}$$

From $\sum_{M \in \mathcal{M}_2^C} P_s^1(M) (\text{Res}(s_{(M,a_1)}, T_1) - \text{Res}(t_{(M,a_1)}, T_1)) = 0$ we obtain $x_{00} + x_{01} = 0$, and from $\sum_{M \in \mathcal{M}_2} P_s^1(M) (\text{Res}(s_{(M,a_1)}, T_1) - \text{Res}(t_{(M,a_1)}, T_1)) = 0$ we obtain $x_{10} + x_{11} = 0$. Similarly, from $\sum_{M \in \mathcal{M}_1} P_s^1(M) (\text{Res}(s_{(M,a_1)}, T_1) - \text{Res}(t_{(M,a_1)}, T_1)) = 0$ we obtain that $x_{01} + x_{11} = 0$. Therefore, we have the following system of equations:

$$\begin{cases} \frac{a_1}{a_1} x_{00} + \frac{a_1}{a_1+b_1} x_{01} + \frac{a_1}{a_1+b_2} x_{10} + \frac{a_1}{a_1+b_1+b_2} x_{11} = 0 \\ x_{00} + x_{01} = 0 \\ x_{01} + x_{11} = 0 \\ x_{10} + x_{11} = 0. \end{cases}$$

The main matrix of the system is

$$\mathbf{Q}_2 = \begin{pmatrix} \frac{a_1}{a_1} & \frac{a_1}{a_1+b_1} & \frac{a_1}{a_1+b_2} & \frac{a_1}{a_1+b_1+b_2} \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

By Lemma 1, $\text{Det}(\mathbf{Q}_2)$ is not a zero-function, which implies that the vector of zero-functions is the only solution of the above system of equations.

We now present how each matrix \mathbf{Q}_{n+1} can be obtained from the matrix \mathbf{Q}_n . In general, for $\mathcal{M}_i^* \in \{\mathcal{M}_i, \mathcal{M}_i^C\}$, it holds

$$\begin{aligned} & \sum_{M \in (\cap_{i=1}^n \mathcal{M}_i^*) \cap \mathcal{M}_{n+1}} P_s^1(M) (\text{Res}(s_{(M,a_1)}, T_1) - \text{Res}(t_{(M,a_1)}, T_1)) \\ & + \sum_{M \in (\cap_{i=1}^n \mathcal{M}_i^*) \cap \mathcal{M}_{n+1}^C} P_s^1(M) (\text{Res}(s_{(M,a_1)}, T_1) - \text{Res}(t_{(M,a_1)}, T_1)) \\ & = \sum_{M \in (\cap_{i=1}^n \mathcal{M}_i^*)} P_s^1(M) (\text{Res}(s_{(M,a_1)}, T_1) - \text{Res}(t_{(M,a_1)}, T_1)). \end{aligned} \quad (21)$$

This means that, in the general case, each solution $x_{i_1 i_2 \dots i_n} = 0$ of the system $\mathbf{Q}_n \mathbf{x} = \mathbf{0}$ generates the following equations for the next system:

$$x_{i_1 i_2 \dots i_k 0 i_{k+1} \dots i_n} + x_{i_1 i_2 \dots i_k 1 i_{k+1} \dots i_n} = 0,$$

for every $0 \leq k \leq n$. For example, in case $|J'| = 3$ we obtain the following matrix:

$$\mathbf{Q}_3 = \begin{pmatrix} \frac{a_1}{a_1} & \frac{a_1}{a_1+b_1} & \frac{a_1}{a_1+b_2} & \frac{a_1}{a_1+b_1+b_2} & \frac{a_1}{a_1+b_3} & \frac{a_1}{a_1+b_1+b_3} & \frac{a_1}{a_1+b_2+b_3} & \frac{a_1}{a_1+b_1+b_2+b_3} \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Note that each row of \mathbf{Q}_3 , except the first one, contains exactly two 1's, at positions whose binary representations differ in exactly one place (for example at the positions 001 and 011).

Informally, the general algorithm for obtaining the elements q_{n+1}^{ij} of a $2^{n+1} \times 2^{n+1}$ matrix \mathbf{Q}_{n+1} from matrix \mathbf{Q}_n , assuming \mathbf{Q}_n is non-singular, is as follows. First, initialize all elements of \mathbf{Q}_{n+1} to zero. Then, copy \mathbf{Q}_n into the upper left corner of \mathbf{Q}_{n+1} . Then, copy \mathbf{Q}_n , excluding the first row, into the lower right corner of \mathbf{Q}_{n+1} . Then, assign 1 to q_{n+1}^{ij} for $i = 2^n + 1$ and $j \in \{2^n, 2^{n+1}\}$.

Finally, add the appropriate new rational fractions in the second half of the first row of \mathbf{Q}_{n+1} . The key observation is that in this way, we obtain again a matrix such that each row, except the first one, contains exactly two 1's, at positions whose binary representations differ in exactly one place. Formally,

$$q_{n+1}^{ij} = \begin{cases} q_n^{ij} & \text{if } 1 \leq i \leq 2^n \text{ and } j \leq 2^n, \\ 1 & \text{if } i = 2^n + 1 \text{ and } j \in \{2^n, 2^{n+1}\}, \\ q_n^{ij} & \text{if } 2^n + 1 < i \text{ and } 2^n < j, \\ \frac{a_1}{a_1 + \sum_{k \in K} b_k + b_{n+1}} & \text{if } i = 1, j > 2^n, \text{ and } q_n^{(i)(j-2^n)} = \frac{a_1}{a_1 + \sum_{k \in K} b_k} \\ 0 & \text{otherwise.} \end{cases}$$

Assuming matrix \mathbf{Q}_n satisfies the conditions of Lemma 1, it easily follows that matrix \mathbf{Q}_{n+1} also satisfies the conditions of Lemma 1. Therefore, its determinant is not a zero function. This means that the system $\mathbf{Q}_{n+1}\mathbf{x} = \mathbf{0}$ has only zero-functions as solutions, which we were aiming to prove. Therefore, the proof of the theorem is complete.

From Theorems 4 and 5 the following statements directly follow.

Corollary 1. *For arbitrary processes s and t , $s \approx_{\mathcal{T}} t$ if and only if $s \approx_{\mathcal{O}} t$.*

Corollary 2. *For arbitrary processes s and t , $s \not\approx_{\mathcal{T}} t$ if and only if there exists a test T without probabilistic transitions such that $\text{Res}(s, T) \neq \text{Res}(t, T)$.*

7 Conclusion, future work, and related work

Concluding remarks We have proposed a testing equivalence in the style of [6] for processes where the internal nondeterminism is quantified with probabilities. The testing semantics allows distribution of external choice over probabilistic choice, i.e. accomplishes unobservability of the internal probabilistic choice. The definition exploits a new method for labeling the synchronized actions using rational functions over the action labels, which, we believe, is of independent interest. We have also developed an alternative characterization of the testing equivalence, namely as a probabilistic version of the ready trace equivalence [1, 21]. The definition of the latter uses Bayesian probability. It is intuitive and can be easily justified by a black box testing scenario akin to those in [4, 10]. We have also shown that it is congruence for all standard operators for the given model, including asynchronous parallel composition and priority.

Internal nondeterminism It can be anticipated by now that combining internal choice, probabilistic choice and parallel composition is challenging. Again “cloning” the internal nondeterminism after the probabilistic choice in a parallel context can “erase” the probabilities, which disallows distribution of prefix over probabilistic choice (this phenomenon has been also studied in [3, 5, 8, 9, 16, 22]). Namely, consider the following game. The player X tosses a fair coin and hides

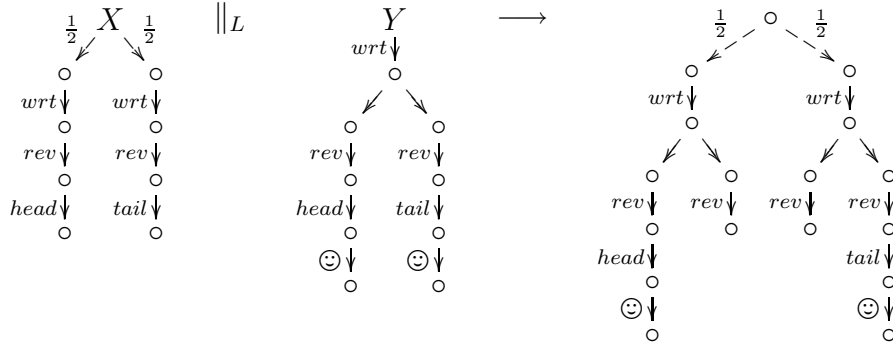


Fig. 5: Synchronized coin tosser(X) and result-guesser(Y)

the outcome. Player Y guesses the outcome of the tossing and writes it down. While he is writing down the result, player X waits (i.e. he may write down something meaningless). Then they both agree to reveal their outcomes, i.e. X to uncover the coin and Y to show what he/she has written.⁵ Obviously, the probability that the second player has guessed correctly equals $\frac{1}{2}$. However, the resulting graph for the synchronization of both players (Fig. 5) suggests that there is a strategy such that player Y can always guess the correct result. On the other hand, if process $\bar{X} = wrt.rev.(head \oplus_{\frac{1}{2}} tail)$ is synchronized with Y , the resulting graph suggests that the probability of reporting a \odot action is exactly $\frac{1}{2}$. This prevents equating processes X and \bar{X} , i.e. allowing distribution of prefix over internal probabilistic choice. Indeed, in presence of internal nondeterminism, the testing equivalence of [25] and its variants have all been characterized as simulations [7, 12, 17]. The proposed solutions [3, 5, 8, 9] to the problem with parallel composition suggest that the process composition needs to “remember” the outcome of the internal choice that a component makes locally. To solve the problem in our setting in the lines of these solutions, we also plan to enrich the internal transitions with labels that cannot communicate. Before composing all labels would be different. If the original process has, for example, two outgoing internal transitions labeled with l_1 and l_2 , then the composed process shall have transitions labeled with $\frac{l_1}{l_1+l_2}$ and $\frac{l_2}{l_1+l_2}$. Fig. 6 presents the result of testing process X of Fig. 5 with process Y , assuming the internal transitions of Y are labeled with l_1 and l_2 . Two processes would not be distinguished by a test if both results of testing are equal modulo isomorphism on the labels set. However, we leave the formal definition of this testing semantics for future work.

⁵ Note the difference between this game and the example in Sec. 1 – in the former there is no external choice in the original processes, while in the latter they don’t have internal nondeterminism.

Related Work Process equivalences that allow distribution of prefix over probabilistic choice (i.e. unobservability of the random choice) have been a research topic ever since probabilities were introduced in concurrency theory (see e.g. [2–4, 13, 16, 18, 22, 24]). However, only [16], [24], and, under certain conditions, [3], also allow distribution of external choice over probabilistic, i.e. equate processes s and \bar{s} of Fig. 1. In [16] probabilistic versions of broom (ready/failure) and barbed (ready/failure trace) equivalences are defined. These definitions use “probability functions” that compute the maximal probability for a

ready trace to occur (i.e. they do not generate probability spaces over the set of ready traces), which makes it hard to construct corresponding “black-box” testing scenarios. In [24], in the model with external choice, a process is defined as conditional probability measure over sequences of actions. This semantics also identifies processes $(a + b) \oplus_{\frac{1}{2}} c$ and $(a + c) \oplus_{\frac{1}{2}} b$. Obviously, this is not desirable. In [3] processes are enriched with labels, and a testing equivalence is defined by means of schedulers that synchronize with processes on the process labels. For a certain labeling, processes s and \bar{s} can be equated. Although this is an elegant and compositional solution to the problem of overestimating probabilities in testing semantics, we believe that our approach is more feasible in practice. In fact, the task of the schedulers and the purpose of the process labels in [3] in our testing semantics have been accomplished by the rational functions formed from the action labels.

Acknowledgements We thank Jos Baeten and Erik de Vink for their valuable comments on a draft version of this paper.

References

1. J. C. M. Baeten, J. A. Bergstra, and J. W. Klop. Ready-trace semantics for concrete process algebra with the priority operator. *The Computer Journal*, 30(6):498–506, 1987.
2. D. Cazorla, F. Cuartero, V. Valero, F. L. Pelayo, and J. J. Pardo. Algebraic theory of probabilistic and nondeterministic processes. *Journal of Logic and Algebraic Programming*, 55(1-2):57–103, 2003.
3. K. Chatzikokolakis and C. Palamidessi. Making random choices invisible to the scheduler. In *Proc. CONCUR 2007*, volume 4703 of *LNCS*, pages 42–58, 2007.
4. L. Cheug, M. I. A. Stoelinga, and F. W. Vaandrager. A testing scenario for probabilistic processes. *Journal of the ACM*, 54(6):29:1–29:45, 2007.
5. L. Cheung, N. Lynch, R. Segala, and F. Vaandrager. Switched PIOA: Parallel composition via distributed scheduling. *Theoret. Comp. Science*, 365(1-2):83–108, 2006.

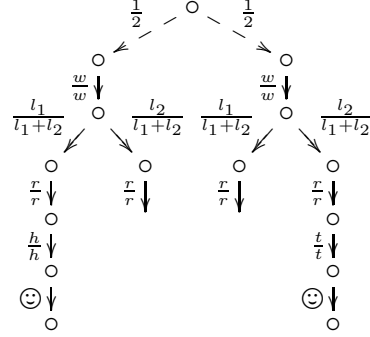


Fig. 6: Testing with internal transitions.

6. R. De Nicola and M.C.B. Hennessy. Testing equivalences for processes. *Theoret. Comp. Science*, 34:83–133, 1984.
7. Y. Deng, R. J. van Glabbeek, M. Hennessy, and C. Morgan. Characterising testing preorders for finite probabilistic processes. *Logical Methods in Computer Science*, 4(4:4):1–33, 2008.
8. F. D. Garcia, P. van Rossum, and A. Sokolova. Probabilistic anonymity and admissible schedulers. *CoRR*, abs/0706.1019, 2007.
9. S. Giro and P. D’Argenio. On the expressive power of schedulers in distributed probabilistic systems. In *Proc. QAPL’09, ENTCS* (to appear), 2009.
10. R.J. van Glabbeek. The linear time – branching time spectrum I; The semantics of concrete, sequential processes. In *Handbook of Process Algebra*, chapter 1, pages 3–99. Elsevier, 2001.
11. C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
12. B. Jonsson and Y. Wang. Testing preorders for probabilistic processes can be characterized by simulations. *Theoret. Comp. Science*, 282(1):33–51, 2002.
13. M. Kwiatkowska and G. Norman. A testing equivalence for reactive probabilistic processes. volume 16(2) of *ENTCS*, 1998.
14. K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.
15. D. V. Lindley. *Introduction to Probability and Statistics from a Bayesian Viewpoint*. Cambridge University Press, 1980.
16. G. Lowe. Representing nondeterministic and probabilistic behaviour in reactive processes. Technical Report PRG-TR-11-93, Oxford Univ. Comp. Labs, 1993.
17. N. Lynch, R. Segala, and F. Vaandrager. Observing branching structure through probabilistic contexts. *SIAM J. Comput.*, 37(4):977–1013, 2007.
18. C. Morgan, A. McIver, K. Seidel, and J. W. Sanders. Refinement-oriented probability for CSP. *Formal Aspects of Computing*, 8(6):617–647, 1996.
19. R. De Nicola. Extensional equivalences for transition systems. *Acta Informatica*, 24(2):211–237, 1987.
20. M. C. Palmeri, R. De Nicola, and M. Massink. Basic observables for probabilistic may testing. In *QEST ’07*, pages 189–200. IEEE, 2007.
21. A. Pnueli. Linear and branching structures in the semantics and logics of reactive systems. In *ICALP’85*, volume 194 of *LNCS*, pages 15–32, 1985.
22. R. Segala. *Modeling and verification of randomized distributed real-time systems*. PhD thesis, MIT, 1995.
23. R. Segala. Testing probabilistic automata. In *Proc. CONCUR ’96*, volume 1119 of *LNCS*, pages 299–314, 1996.
24. K. Seidel. Probabilistic communicating processes. *Theoret. Comp. Science*, 152:219–249, 1995.
25. Y. Wang and K. G. Larsen. Testing probabilistic and nondeterministic processes. In *Proceedings of the IFIP TC6/WG6.1 Twelfth International Symposium on Protocol Specification, Testing and Verification XII*, pages 47–61, 1992.